

REMARKS

In the Official Action mailed on **22 December 2005** the Examiner reviewed claims 1-3, 5-7, 11-13, 15, 16, 20-24, 28, 29 and 31-34. Claims 1-3, 5-7, 11-13, 15, 16, 20-24, 28, 29 and 31-34 were rejected under 35 U.S.C. §102(b) as being anticipated by Rothrock (USPN 5,408,470, hereinafter "Rothrock"). Claims 1, 2, 5, 7, 11-13, 15, 20-22, 24, 28, 29 and 31-34 were rejected under 35 U.S.C. §102(e) as being anticipated by Zhu (USPN 6,792,436, hereinafter "Zhu"). Claims 3 and 23 were rejected under 35 U.S.C. §103(a) as being unpatentable over Rothrock. Claims 6 and 16 were rejected under 35 U.S.C. §103(a) as being unpatentable over Rothrock in view of Zhu.

Rejections under 35 U.S.C. §102(b) and §102(e)

Independent claims 1, 12, 21, 28, and 32-34 were rejected as being anticipated by Rothrock and Zhu.

Applicant respectfully points out that the present invention is directed towards synchronizing database caches (see page 1, lines 6-8; page 7, lines 1-30 of the instant application). Specifically, the present invention synchronizes two copies of an object by (a) determining an object change set at a first system, (b) sending the object change set to a second system, and (c) **returning control of the object persistence system in the first system to a client coupled to the first system whether or not the second system has completed processing the object change set**, and (d) applying the object change set to the corresponding object on the second system to synchronize the object on the second system with the object on the first system (see page 7, lines 1-30; page 13, lines 23-29; and page 14, lines 1-12 of the instant application).

The benefit of returning control of the object persistence system in the first system to a client coupled to the first system whether or not the second system has completed processing the object change set is that the client can immediately

continue making changes to objects in the first system without waiting for the second system to complete processing the object change set.

Applicant respectfully points out that Rothrock teaches synchronizing a list of objects between a first system and a second system by: (a) modifying the object on the first system, (b) requesting an object index from an arbitrator, (c) receiving an object index from the arbitrator, (d) synchronizing the first system with the arbitrator, (e) broadcasting changes to a second system (see Rothrock column 10, lines 12-25; column 10, lines 45-64; column 11, lines 16-21; column 11, lines 50-67; column 12, lines 10-37).

Note that although, the first system in Rothrock is allowed to *continue making changes to objects in the time period between requesting an **object index** from the arbitrator and receiving an object index from the arbitrator* (see Rothrock column 10, lines 45-64; column 11, lines 56-62; column 12, lines 10-13), these changes are allowed only **prior to computing and sending** the object change set to the second system. There is no suggestion, either implicit or explicit, which suggests that **after sending the object change** set to the second system, the first system is allowed to continue making changes to objects in the first system whether or not the second system has completed processing the object change set.

In contrast, the present invention computes an object change set **after changes for a given transaction have been committed** in the first system (see page 7, lines 5-10 of the instant application). After sending the change set to the second system, a client coupled to the first system **regains control** of the object persistence system in the first system whether or not the second system has completed processing the object change set. Hence, a client coupled to the first system can begin modifying objects in the first system immediately after sending the object change set to the second system without having to wait for the second system to complete processing the object change set.

There is nothing in Rothrock of Zhu, either implicit or explicit, which suggests that after sending the object change set from the first system to the second system, a client coupled to the first system regains control of the object persistence system in the first system whether or not the second system has completed processing the object change set.

Accordingly, Applicant has amended independent claims 1, 12, 21, 28, and 32-34 to clarify that after sending the object change set to from the first system to the second system, **control is returned to a client coupled to the first system whether or not the second system has completed processing the object change set**. These amendments are supported on page 7, lines 1-30; page 13, lines 23-29; and page 14, lines 1-12 of the instant application.

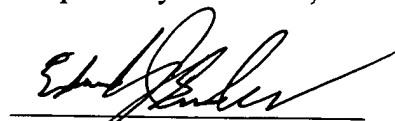
Hence, Applicant respectfully submits that independent claims 1, 12, 21, 28, and 32-34 as presently amended are in condition for allowance. Applicant also submits that claims 2-3, 5-7 and 11, which depend upon claim 1, claims 13, 15-16 and 20, which depend upon claim 12, claims 22-24, which depend upon claim 21, and claims 29 and 31, which depend upon claim 28, are for the same reasons in condition for allowance and for reasons of the unique combinations recited in such claims.

CONCLUSION

It is submitted that the present application is presently in form for allowance. Such action is respectfully requested.

Respectfully submitted,

By



Edward J. Grundler
Registration No. 47,615

Date: 8 February 2006

Edward J. Grundler
PARK, VAUGHAN & FLEMING LLP
2820 Fifth Street
Davis, CA 95616-7759
Tel: (530) 759-1663
FAX: (530) 759-1665
Email: edward@parklegal.com